ONLINE EDGE COLOURING

Shridhara Pradhan
Computer Science & Engineering
VSSUT
Burla, Sambalpur
shree55lenovo@gmail.com

1603041024

ABSTRACT- Edge colouring can be used to solve problems in all discipline. This paper considers the problems of online edge colouring. Here the graphs are taken randomly and must be assigned a colour where all the edges that are adjacent to each other must be given different colours. In otherwords all the edges incident to any specific vertex must contain no repeated colours. There are two possibility of colouring a graph i.e. when the whole graph is known in advance and when the graph is unknown. There are some algorithms which solve the problem of graph colouring. Some are offline algorithm & some are online algorithm .Where offline means the graphs is known in advance and the online means that edges of the graph are arrive one by one as an input. We need to colour each edge as soon as it is added to the graph and also we have to check that the number of colours is less. We cannot change the colour of an edge after coloured in an online algorithm. The minimum number of colours is called as the chromatic number of the graph. The applications of edge colouring mainly include scheduling and minimizing the number of rounds or time slots needed to complete all tasks.

Keywords- *First Fit, Backtracking, Chromatic Number, Vizing's Theorem*

I. INTRODUCTION

Graph colouring is one of the most important concepts in graph theory and is used in many real time applications in various fields. Various colouring methods are available and can be used on requirement basis. The proper colouring of a graph is the colouring of the vertices and edges with minimal number of colours such that no two vertices should have the same colour. The minimum number of colours is called as the chromatic number and the graph is called properly coloured graph. We need to colour each edge as soon as it is added to the graph and the main issue is that we want to minimize the number of colours. We cannot change the colour of an edge after coloured in an online algorithm. In this paper, we improve the online algorithm for edge colouring. There is also a theorem which proves that if the maximum degree of a graph is Δ, then it is possible to colour its edges, in polynomial time, using at most Δ+ 1 colour. The algorithm provided by Vizing is offline, i.e., it assumes the whole graph is known in advance.

Edge Colouring: Edge colouring is a problem in graph theory where all the edges in a given graph must be assigned a colour. Furthermore, all edges that are adjacent to each other must be given different colours. In other words, all the edges incident to any specific vertex must contain no repeated colours.

The goal of this problem is to minimize the number of colours needed to colour all the edges in the graph legally. It would be trivial to assign a different colour to every edge in the graph.

If G is k-edge colourable but not (K-1) edge colourable , we say that the chromatic index of G , $\mathbf{X'(G)=K}$.

Since edge colouring an end vertex need different colours.

$$\mathbf{X'(G) >= \Delta\ (G)}$$

Vizing's Theorem: Before going into specifics of edge colouring algorithms, it is necessary to define a couple of key terms and introduce Vizing's Theorem, which is essential to the study of edge colouring.

$\chi'(G)$: This is referred to as the edge chromatic number or chromatic index, which is the minimum number of colours needed to edge colour graph G.

.Note that this term is different from $\chi(G)$, which is simply called the chromatic number and defines the minimum number of colours to vertex colour graph G. Vertex colouring is a different type of graph problem but the relationship between edge colouring and vertex colouring will be shown later.

$\Delta(G)$: This number, delta, is the maximum degree of graph G. That is, within the set of vertices in G, Δ represents the vertex (or vertices) with the highest number of incident edges within the graph.

There is a close relationship between the maximum degree of a graph and its edge chromatic number.

Vizing's Theorem states that for any simple graph, the edge chromatic number is either Δ or Δ + 1. For multi graphs, a loose upper bound was shown by Shannon to be (3Δ/2). This bound will be explained in more detail in the multi graph edge colouring section to fit Vizing's Theorem.

In mathematical terms:

$\mathbf{\Delta(G) \leq \chi'(G) \leq \Delta(G) + 1}$, for any simple graph G.

A Class 1 graph is defined as a graph G where $\chi'(G) = \Delta(G)$. Bipartite graphs and high degree planar graphs are all of Class 1. Any graph not in Class 1 is considered a Class 2 graph.

While most random graphs are Class 1, it is an NP-hard problem to determine whether a graph is of Class 1 or Class 2. To restate Vizing's Theorem, all simple graphs can be edge coloured with either Δ or Δ + 1 colours. The goal of a good edge colouring algorithm is one that guarantees using at most Δ + 1 colours.

Class II if $\chi'(G)=\Delta(G)$

Class II if $\chi'(G)=\Delta(G) + 1$

Greedy algorithm: It is an algorithm in which we can choose the maximum value or the minimum value for choosing next value it will follow descending or ascending order respectively.

Backtracking: This process is widely used in various algorithms. When we try to solve a problem according to any procedure at that time we assumes that we will get the correct result we will get perfect output but in later time when we reach at the last state but did not get the output as we expect from the beginning of that work, hence at that time move backward towards the starting point through the same path in which we came to find out the flaws or mistakes we have done before and whenever we get any

1603041024

flaws or mistakes then we will rectify that and this called backtracking.

II. APPLICATION OF ONLINE VERTEX COLOURING

- Making schedule or Time table
- Mobile radio frequency assignment
- Sudoku
- Register allocation
- Map colouring
- Resource allocation
- Routing

III. WORK DONE

PROPOSED ALGORITHM

1. Choose the first Edge randomly.
2. Assign a colour code-1 to all the Edges.
3. Repeat step (4) and step (5) for all Edges.
4. Check the adjacent edges of chosen Edge.
5. If adjacent edge having same colour as current chosen edge then increment the colour code and go to step (4).
6. Chromatic Number = Maximum colour code
7. Exit.

IMPLEMENTATION:

This algorithm I have implemented in 10 types of graphs and those are as follows:

1. Forest graph
2. Simple graph
3. Bipartite graph
4. Tripartite graph
5. Peterson graph
6. Regular graph
7. Cyclic graph
8. Planar graph
9. Star graph
10. Wheel graph

Forest graph: Let G be a graph with V vertices and E edges i.e. G= (V, E). We can say G is a forest graph if V=n and E=0. Its chromatic number is 1.

Simple graph: Let G be a graph with V vertices and E edges i.e. G= (V, E). If a graph G does not have any self-loop or any parallel edges. Then it will be a simple graph. If in a simple graph G total number of vertices is N, then degree of each vertices and the number of edges are at most (N-1) and (N*(N-1)/2) respectively.

Bipartite graph: Let G be a graph with V vertices and E edges i.e. G= (V, E). If a graph G can be divided into 2 subsets in such a way that every vertices of subset1 should not be adjacent to any of its vertex, and similar for subset2. And there should be at least one edge between two subsets. Let the graph is G and the 2 subsets are G1 & G2 respectively. All the vertices of G1 should not be adjacent to any of its vertex and same for G2. And there should be at least one edge present between G1 & G2. If in a bipartite graph G total number of vertices is V, and subset G1 has vertices = V1, and G2 has vertices = V2. So the number of edges at most is = (V1*V2). Chromatic number is 2.

Tripartite graph: Let G be a graph with V vertices and E edges i.e. G= (V, E). If a graph G can be divided into 3 subsets in such a way that every vertices of subset1 should not be adjacent to any of its vertex, and similar for subset2 and subset3. And there should be at least one edge between subset1, subset2 and subset3. Let the G is a tripartite graph and the three subsets are G1, G2 and G3 respectively. All the vertices of G1 should not be adjacent to any of its vertex and same for G2 and G3 too. And there should be at least one edge present between (G1 & G2), (G2 & G3), (G3 & G1). If in a tripartite graph G total number of vertices is V, and subset G1 has vertices = V1, subset G2 has vertices = V2 and subset G3 has vertices = V3. So the number of maximum edges can be present in that tripartite graph = ((V1*V2) + (V2*V3) + (V3*V1)) Chromatic number is 3.

1603041024

Peterson graph: Let G be a graph with V vertices and E edges i.e. G= (V, E). If a graph G has 10 vertices and 15 edges and its chromatic number is 3 then it will be a Peterson graph. So here the number of vertex is (V) =10. The number of edges is (E) =15. The chromatic number is = 3

Regular graph: Let G be a graph with V vertices and E edges i.e. G= (V, E). If a graph G has N number of vertices, and all the vertices has same degree. The chromatic number is dynamic.

Cyclic graph: Let G be a graph with V vertices and E edges i.e. G= (V, E), it is basically similar to a line graph but in this graph both the end vertices have an edge between them by which it forms a cycle. Here if the number of vertices is odd, then the chromatic number is 3 otherwise 2.

Planar graph: Let G be a graph with V vertices and E edges i.e. G= (V, E). If a graph G has N number of vertices, if a graph can be drawn in plane in such a way that no vertices should cross each other then it will be known as planar graph. The chromatic number is 4.

Star graph: Let G be a graph with V vertices and E edges i.e. G= (V, E). If a graph G has N number of vertices, then a vertex has (N-1) degree and all the other vertices will have degree 1. The chromatic number is 2.

Wheel graph: Let G be a graph with V vertices and E edges i.e. G= (V, E). If a graph G has N number of vertices, then a vertex has (N-1) degree and all the other vertices will have degree 2. Here if the number of vertices is odd, then the chromatic number is 3 otherwise 4.

Dry run:

SI NO.	***GRAPH NAME***	***CHROMATIC NUMBER***
1	Forest graph	1
2	Simple graph	Dynamic
3	Bipartite graph	2
4	Tripartite graph	3
5	Peterson graph	3
6	Regular graph	Dynamic
7	Cyclic graph	3(odd), 4(even)
8	Planar graph	4
9	Star graph	2
10	Wheel graph	3(odd), 4(even)

IV. FUTURE WORK

As with any algorithmic analysis of a problem, the ultimate goal is to find the best possible algorithm that will solve the problem. We can stop once we have found a tight lower bound for *any* algorithm solving the problem that matches the upper bound determined by a well-defined algorithm. As the above discussion suggests, we are not finished with the online graph colouring problem.

Also we have plan to implement **Next Fit Algorithm** on Some Standardized graph to see whether the graph can be coloured or not.

V. CONCLUSION

The online graph colouring problem has been studied extensively over many years, but the analysis of the problem is not complete. General lower bounds of Ω(n/(log n) 2) have been proven for the deterministic and randomized models of computation, however these do not match the current upper bounds that have been found. It is also important to note that online graph colouring does not seem to get easier after relaxing some of the restrictions placed on the algorithm by the problem. Several algorithms have been proposed to solve online graph colouring, each successful in their own way.

The First Fit algorithm is the simplest and most intuitive, and also tends to work very well for a variety of specific graph classes, for graphs with bounded path width, and is effective in the asymptotic case. The weak point for the FF algorithm is the disastrous potential performance ratio of n/4

16304024

which arises for certain bipartite graphs. Thus the FF algorithm cannot claim to have sub-linear performance ratio. Much work is left to be done before the book can be closed on the online graph colouring problem. Perhaps the day will come when mankind has fully analysed the online graph colouring problem as well as the other difficult NP-complete problems.

Finally in backtracking method we are the we will get the correct result we will get perfect output but in later time when we reach at the last state but did not get the output as we expect from the beginning of that work, hence at that time move backward towards the starting point through the same path in which we came to find out the flaws or mistakes we have done before and whenever we get any flaws or mistakes then we will rectify that and this called backtracking.
The we can clearly sees that the above graphs can be coloured using first fit and backtracking method.

VI. REFERENCE

1. A 1.43-Competitive Online Graph Edge Coloring Algorithm In The Random order arrival modelI(BahmanBahmani∗Aranyak Mehta† Rajeev Motwani‡).
2. A-study-of-Vertex-Edge-Coloring-Techniques-with-Application.(International Journal Of Core Engineering & Management(IJCEM) Volume 1, Issue 2, May 2014).
3. Edge Coloring Algorithms Research Report by Timothy Ho.
4. Extended Online Graph Edge Coloring By Lata Naharwal1 and DalpatSongara2(International Journal on Computational Sciences & Applications (IJCSA) Vol.4, No.1, February 2014).
5. Online Graph Colouring, Avery Miller, December 6, 2004.
6. https://en.wikipedia.org/wiki/Online_algorithm
7. http://homepages.cwi.nl/~lex/files/edgecol.pdf
8. https://en.wikipedia.org/wiki Edge_coloring.
9. *www.eecs.tufts.edu/~tho01/edgecoloringreport_tho.pdf*

1603041024

1603041024